**INTRODUCTION**

The introductory part of this report is aimed at critically evaluating and comparing the commonly used Data Warehouse Technologies we presently have in the field of cloud computing, and then concludes with a summarization of the evaluation result and also, give possible future development on chosen platforms.

***The technologies I will be evaluating in this research work are as follows*:**

* Google BigQuery
* Microsoft Azure
* Redshift
* Snowflake

***The evaluation criteria are as follows:***

* Performance at scale (the scalability of the technology in terms of query latency and concurrency)
* elasticity
* ease of use (does the technology require infrastructure management)
* cost efficiency
* data freshness (ability to support continuous ingestion of data)
* supports structured and semi-structured data
* data granularity (the granularity level at which the platform operates)
* concurrency (ability of a user to run multiple queries at the same time)
* ecosystem integration (does the platform provide api and sql access)

***Research methodology***

The methodology used include an assessment technique in which each platform was extensively reviewed in light of research and experimentation, with correlations as the report's conclusion.

**THE INVESTIGATION**

**GOOGLE BIGQUERY**

* *Performance at scale:*

It has been noted that consumers will not need to bother about scalability at all because everything is done behind the scenes. Moving your data onto Google Cloud will be the most difficult part of building a petabyte-scale data warehouse. Then, all you have to do is execute your query, and BigQuery will take care of the rest (Google, BigQuery Enterprise Data warehouse, 2019).

* *Ecosystem integration:*

BigQuery integrates with the Apache big data ecosystem via Dataproc, a fully managed and highly scalable service, allowing current Hadoop/Spark and Beam load to read or write data directly from BigQuery via the API Storage; it also supports over 30 tools and frameworks, SQL included.

* *Cost efficiency:*

In terms of inquiries, there are two price levels: on-demand pricing and pricing depending on the amount of data handled by each query. Pricing is determined on the amount of storage and query processing volume. Storage is priced on two levels. For data that has been updated within the past 90 days, active storage is paid per month. Data that has not been modified in 90 days is charged at a cheaper monthly rate.

* *Ease of use:*

Users focus less on infrastructure, performance, security, elasticity, and dependability while using BigQuery's serverless and fully managed approach, and more on generating insights from data. It is a Platform as a Service (PaaS) that supports ANSI SQL querying. BigQuery employs a proprietary format so that it may expand in collaboration with the query engine, which leverages deep knowledge of the data layout to enhance query performance (Google, BigQuery Enterprise Data warehouse, 2019).

* *Supports structured and semi-structured data:*

It provides over 100 connections to prominent data sources such as Google Analytics, Google Ads, and Facebook Ads. Stitch makes the work of loading BigQuery quick and easy by supporting the storage and processing of both structured and semi-structured data such as JSON at scale.

* *Concurrency:*

You can conduct up to 10 concurrent queries using remote functions. Your project can handle up to 1,000 simultaneous multi-statement queries. Your project can perform up to six classic SQL queries with user-defined functions at the same time. This restriction applies to interactive and batch queries. Your project may perform up to 100 interactive queries at the same time (Google, BigQuery Enterprise Data warehouse, 2019).

* *Granularity:*

Big Query splitting produces smaller portions of a single table (partitions) based on particular criteria. Big Query, for example, may build a new table containing only the data (daily, hourly, monthly, or yearly partitioning). Tables in Big query may be partitioned into Time-unit column, Ingestion time, and Integer range.

* *Data freshness:*

BigQuery may be used to ingest, analyze, store, and even do basic analytics on data. If you have an external data source, you can either load it into BigQuery or query it without importing it at all (Federated Query). It even allows real-time data input at millions of times per second.

**MICROSOFT AZURE**

* *Performance at scale:*

Azure Auto scale services can grow automatically to handle load and fulfill demand. Azure Edge Zones and 5G networks' both high bandwidth and little edge latency can simply deploy apps and virtualized network functions (VNF) and give your customers seamless computing, storage, IoT, and container services (Microsoft, 2022).

* *Ecosystem integration:*

For both on-premises and online programmes, as well as for data and processes, Microsoft Azure has four kinds of integration technologies: Service Bus, API Management, Event Grid, and Logic Apps (Zhu, 2022).

* *Cost efficiency:*

The price of Azure SQL data warehouse is divided into two sections of computation fee and storage fee. When you pause, you will just be charged for storage rather than compute. There are no up-front or cancellation fees (Microsoft Azure, 2022).

* *Ease of use:*

Azure uses virtualization technology, with cloud server(racks) set up in one or more Data centers to clusters, to run user-facing virtualized hardware instances. Some servers, however, utilize a fabric controller which is a cloud management tool. These applications control the installation and use of virtualized hardware and software on the Azure server computers. Because Azure takes care of this for users, they don't have to maintain or upgrade their hardware when utilizing it.(Microsoft Azure, 2022).

* *Supports structured and semi-structured data:*

Azure Cognitive Search may use a semi-structured data indexer to index JSON documents and arrays in Azure Blob Storage. Using the syntax column-name>: you extract a column from fields holding JSON strings. <extraction-path>

* *Concurrency:*

A single function app can only support 200 maximum occurrences. However, a single instance may handle several messages or requests concurrently, therefore there isn't a hard restriction on number of concurrent executions.

* *Granularity:*

Azure Data Factory operates on Azure infrastructure which incurs fees as you launch new materials. It is critical to note that additional infrastructure expenditures may arise. For instance, bandwidth expenses will be incurred while transferring data across availability zones. The reporting on pipeline billing won't contain these costs (Zhu, 2022).

* *Data freshness:*

By using ingestion wizard speeds up the process of ingesting data, establishing database tables, and mapping structures. Choose data from several sources in various data formats, either as a one-time or continual ingestion procedure (Microsoft Azure, 2022).

**AMAZON REDSHIFT**

* *Performance at scale:*

Scale quickly with Amazon Redshift in three ways: Resize your Redshift clusters in a couple of hours by using Redshift Spectrum to query data in your Amazon S3 data lakes without putting it into the cluster by adding more nodes or changing node types. Analysts can continue to conduct read queries without interruption throughout this time or build up numerous Amazon Redshift clusters by quickly recovering data from a snapshot (Amazon, 2019).

* *Ecosystem integration:*

Python, Go, Java, Node.js, PHP, Ruby, and C++ are just a few of the programming languages and platforms that can access, consume, and export data using the Redshift Data API. Driver configuration and database connection management are not needed thanks to the Data API. As an alternative, you can use a secure API endpoint made available via the Data API to conduct SQL queries on an Amazon Redshift cluster.

* *Cost efficiency:*

Starting small and scaling up to petabytes of data and thousands of concurrent users is possible with Amazon Redshift for $0.25 per hour. To save money and get the best performance, you may add and remove nodes using Resize Scheduler on a daily or weekly basis (Amazon, 2019).

* *Ease of use:*

AWS automates the effort required to set up, administer, and extend a data warehouse on your behalf, allowing you to concentrate on developing your applications. Redshift Spectrum's whole computational architecture, load balancing, planning, scheduling, and execution of your queries on Amazon S3 data are all managed by Amazon Redshift. (Amazon, 2019) .

* *Supports structured and semi-structured data:*

Redshift extends data warehouse capacity by integrating with SQL and NoSQL data sources using the PartiQL language and the SUPER data type. Redshift is a technology that offers efficient analytics on relational and semi-structured stored data, including JSON.

* *Concurrency:*

Concurrency Scaling allows you to accommodate almost limitless simultaneous usage of resources and query execution while ensuring continuously fast query execution. In order to manage an increase in both read and write requests, Redshift increases cluster capacity when concurrent scaling is enabled.

* *Granularity:*

You may exchange data at several levels using Amazon Redshift. Examples of these layers include databases, schemas, tables, views (such as ordinary, late-binding, and materialised views), and SQL user-defined functions (UDFs). Fine-grained access control is obtained by having this flexibility in data sharing. This control may be customized for various people and enterprises who want access to Redshift data (Amazon, 2019).

* *Data freshness:*

With Redshift's new streaming ingestion capabilities, you may utilize SQL (Structured Query Language) within Redshift to connect to and directly ingest data from several Kinesis data streams at the same time.

**SNOWFLAKE**

* *Performance at scale:*

Snowflake can perform a nearly infinite number of concurrent workloads against the same, single copy of data due to the separation of storage and computing (Snowflake, 2022).  Vertical scaling implies that the platform provides more computing power to existing virtual warehouses, such as by upgrading CPUs. More cluster nodes are added as a result of horizontal scaling.

* *Ecosystem integration:*

Accessing and modifying data in a Snowflake database is made possible using the Snowflake SQL API, a REST API. This API may be used to develop personalized integrations and applications that manage your deployment and run queries (e.g., provision users and roles, create tables, etc.) It has functions for sending SQL statements for execution, checking the progress of a statement's execution, and stopping the execution of a statement.

* *Cost efficiency:*

Storage price is based on the average monthly terabytes of all Customer Data kept in the Snowflake Account. The average terabytes per month is computed by obtaining an hourly snapshot of all Customer Data and then averaging this across each day. If Customer Data is compressed and saved, the compressed file size is utilized to calculate total storage consumed (Snowflake, 2022).

* *Ease of use:*

Snowflake requires practically no management; you can be up and operating in Snowflake in minutes. As a user, you are not required to see behind the scenes of Snowflake's work. All management, maintenance, updates, and tuning chores are handled by the platform. It also handles all elements of programme installation and maintenance. This is true for all categories of users, including regular end users, business analysts, and data scientists.

* *Supports structured and semi-structured data:*

Snowflake is excellent for semi-structured data and treats it as a first-class database element. Snowflake's design enables SQL queries on semi-structured and structured data simultaneously. In a single query, you may join, window, compare, and compute structured and semi-structured data.

* *Concurrency:*

A Snowflake multi-cluster warehouse is made up of one or more clusters of query servers. A Snowflake customer can specify the minimum and maximum number of compute clusters to allocate to a specific warehouse (Snowflake, 2022).

* *Granularity:*

The degree of access allowed can be managed by a variety of different rights. In the Snowflake paradigm, privileges provided to roles which are then assigned to other roles or users are used to grant access to secure objects. Each secure item has an owner who has the authority to provide access to different roles. This approach differs from user-based access control models, in which each individual user or group of users is given certain rights and privileges. The goal of the Snowflake technique is to offer a great degree of control and flexibility.

* *Data freshness:*

Snowpipe is Snowflake's service for continuous data intake. Snowpipe loads data within minutes after adding files to a stage and submitting them for ingestion. With Snowpipe's serverless computing approach, Snowflake maintains load capacity, assuring appropriate compute resources to meet demand. Snowpipe, in a nutshell, provides a "pipeline" for loading new data in micro-batches as soon as it becomes available (Snowflake, 2022).

**FINAL COMPARISON**

BIGQUERY: intended for typical SQL queries on structured and semi-structured data. It is incredibly cost effective and highly tuned for query performance. BigQuery is a fully managed cloud service, therefore there is no operational overhead. It is better suited for interactive queries and OLAP/BI applications. With the addition of support for unstructured data, Borg, Colossus, and Jupiter are examples of Google's cloud infrastructure technologies that are important differentiators because BigQuery service outperforms some of its competitors (Google, BigQuery Enterprise Data warehouse, 2019).

AZURE: a worldwide network of Microsoft-managed datacenters can be rapidly and easily accessed by using an open and flexible cloud platform that enables you to create, manage, and deploy apps. Any language, tool, or framework may be used to create apps. Additionally, you may combine your present IT infrastructure with the public cloud apps you use. It functions as a platform that also enables you to scale up applications with limitless servers and storage. We'll get virtual machines, rapid data processing, analytical and monitoring tools, and other things to make our jobs simpler. Azure has a straightforward and affordable pricing structure. commonly known as "Pay As You Go."

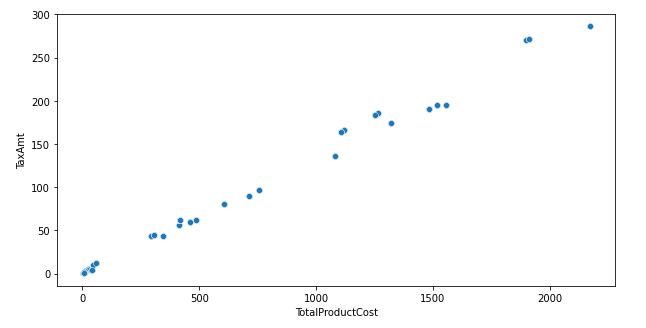
REDSHIFT: supplies entirely handled petabyte-scale storage along with an enterprise-class relational database management system that supports client connections with a range of applications, including reporting, analytical tools, and enhanced business intelligence (BI) applications, where you can query substantial amounts of data in multiple-stage operations to produce final results, all of which are done at very efficient storage and query performance. (AWS, Redshift, 2022).

SNOWFLAKE: provides the speed, parallelism, and simplicity required to centrally store and analyze all data accessible to an organization. With its technology, Snowflake offers real-time data sharing, elasticity of the cloud, and the power of data warehousing for a fraction of the price of competing products. Snowflake: No restrictions on your data (Snowflake, 2022).

**THE BIG DATA WAREHOUSE IMPLEMENTATION REPORT**

**EXPLORING THE DATASET**

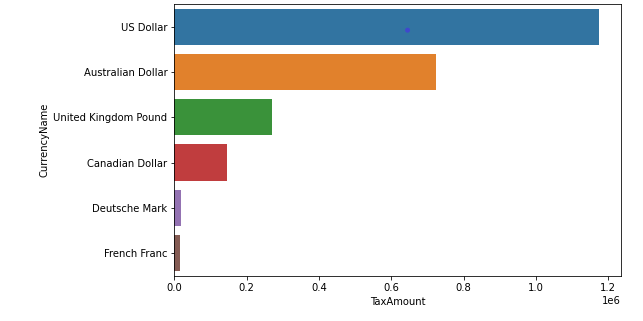
***Scatter plot of Tax amount against Total Product cost***



The relationaship found between Tax amount and Total cost of product is so strong that, the scatter plot is almost a perfect straight line with a correlation of **0.99** which can be approximated to **1.**

This kind of relationship can be said to directly proportinal, i.e the higher the cost of a good, the higher its Tax rate, and vice-versa.

***Sum of Tax against Currency name***

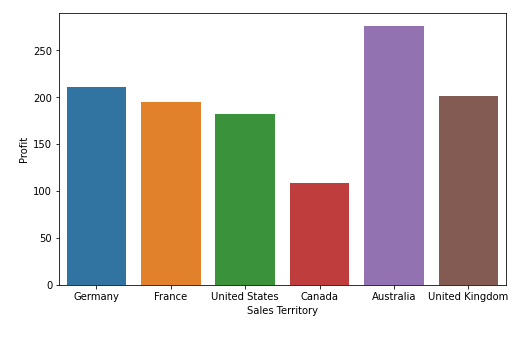


It is necessary to know the taxation rate based on the currencies involved, as seen above, US Dollar has the highest tax amount, followed by Australian Dollar, while the least is the French Franc.

This might be according to the popularity of each currency or the amount of Unit of product bought.

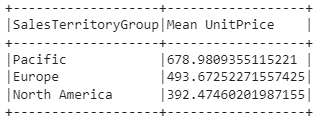
A column for Profit(difference in cost and price) was created to look into the returns of the organisation within the period concerned. I explored the Profit column based on the Sales Territory(Country), and bellow is my finding.

***Average Profit made in each Country***



As seen above, sales Territories in Australia made the mean highest profit of 276, followed by Germany at 211, then United Kingdom with 201.3, down to Canada having the least profit of 109.

***Average Unit Price in each Territory Group***

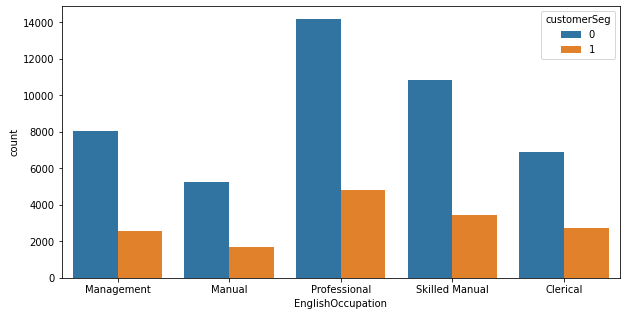


From the above table, it can be seen that the territory group with the highest Unit Price is the Pacific with 678.9, followed by Europe having 493.6, and lastly North America having a value of 392.2

On the fact table, a column was created called TotalSpent, which is the summation of UnitPrice, TaxAmt and Freight. This was done to know the total amount spent in each record.

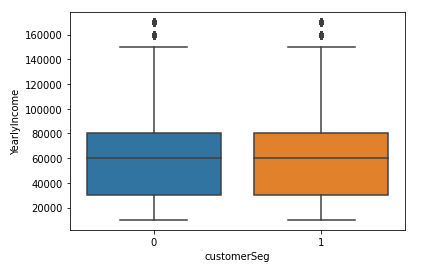
Hypothetically, the higher the earning of a person, the higher the daily expense, this led me to check the relation between the earning of Customers and the total spent on goods.

After finding the correlation between the two columns, the value gotten was **0.07** which is very far from zero, and can as well be approximated as zero.

***Count plot of occupation of good and bad customer*** 

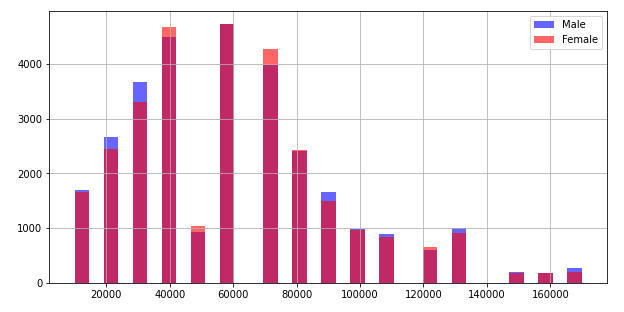
from the newly created column *CustomerSeg,* which represents customer who spent above average and those who didn’t, I made a joint count plot.

***Boxplot of customer segment spending rate spending rate***



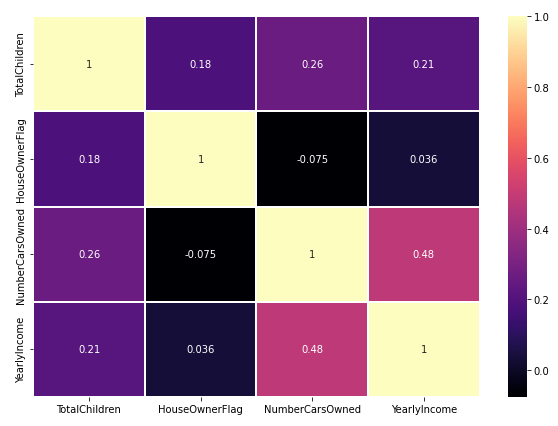
Above is the boxplot depicting the earning rate of both ‘good’ and ‘bad’ customers, as seen, the earning distribution are mirror of each other, hence it is safe to say both categpries of customer earn at the same level, so their earning as no effect on their spending power.

***Yearly income of both genders***



From the TotalSpent column, there is need to check the spending rate between both Genders, and between cutomers who own a house and those that don’t.

***Correlation heatmap***



Above is a correlation heatmap containing TotalChildren, HouseOwnerFlag, NumberCarsOwned and  YearlyIncome columns. There is no strong correlation between any of the variables, only **YearlyIncome and NumberCarsOwned** are having the highest correlation of **0.48**, while others are having nearly-insignificant correlation among them.

**PRICE PREDICTION**

**Aim:** to make a price prediction

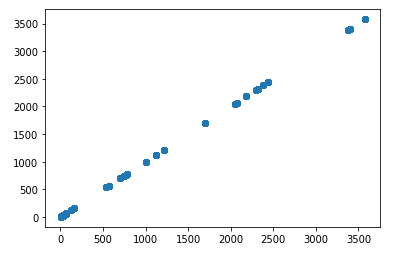
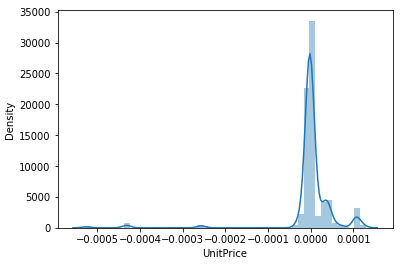
**Methods:** Linear Regression, Polynomial Regression and Random Forest Regression.

During the preprocessing stage, the dataset was splitted into two, Test and Train using a ration 7 to 3 respectively, all necessary model and function were imported and applied.

**LINEAR REGRESSION.**

The intercept of -3.47093e-05 was generated while the corresponding coefficient is as follow:

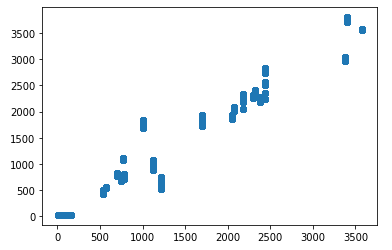
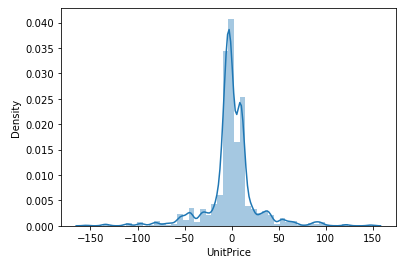


As seen from the plots of **y\_test and prediced** variable above, the scatterplot is on a perfect straight line, which is nearly imporsible, also looking at the histogram, it is skewed to the left, which is an indicator of abnormality in its distribution.

Having an Explained Variance Score of 0.9999999999999951, this can as well be safely approximated to 1, meaning it generated a perfect prediction which is not realistic as far as machine learning prediction is concerned and can lead to overfitting, it can then be said that the dataset is non-linear in nature, hence the need to try another model.

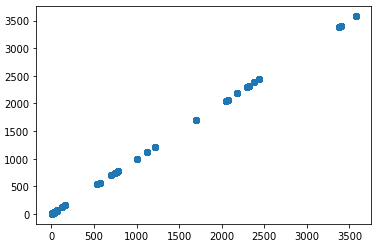
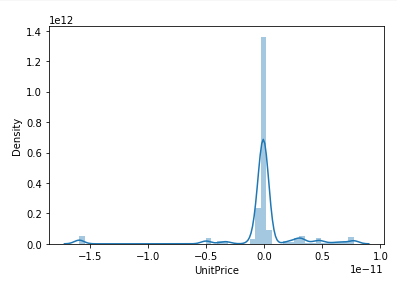
**POLYNOMIAL REGRESSION.**

The results of the Polynomial regression can be seen above, the scartter plot maintains a realistic straigth path, while the histogram of the Residuals also shows a normal distribution to some extent which is an indicator of a good model.

Having an Explained Variance Score of 0.9877101852687572

**RANDOM FOREST REGRESSION**

Equally as the previous algorithms, above afre the scartter plot of the dependent variable against the predicted, and the histogram of the residuals respectfively.

As the linear regression scatterplot, the RFR plot is also too perfect to be accepted, but considering the histogram of its residuals, the distribution looks fair and a bit realistic, which makes the RFR algorithm more aceptable with an **Explained Variance Score** of 0.998371783957064.

Among the three algorithms deployed, the **Polynomial Regression** appears to have the best output using an order of **5,** followed by the **Random Forest Regression** which is not totally bad either, but I wouldn’t recommend **Linear Regression** in this context as the dataset does not appear to be linear in nature, and hence the reason for the unrealistic result of the model.

In a nutshell, the dataset is polynomial in nature, this is determined by the results gotten from the alternate algorithms in comparison to each other.

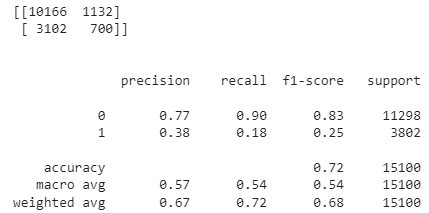
**CLASSIFICATION**

**Aim:** to classify the cutomers into ‘Good’ and ‘Bad’ according to their spending power.

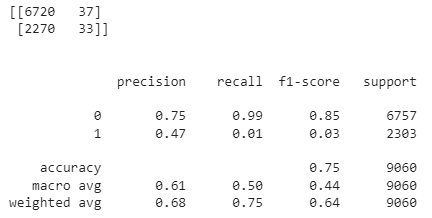
**Methods:** Decision Tree Classifier, Support Vector Machine, and Logistic Regression

A column was created called Total spent, which was the total amount spent by each customer, then from there, I created another categorical column called customer segmentation, wchich contains either a customer spends above or below average: this is the column I made my classificatioon prediction on

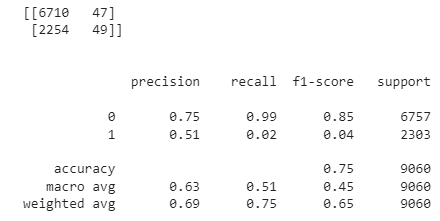
**NAIV BAYES THEOREM:** chosen for its simplicity and versatility



**SUPPORT VECTOR MACHINE**



**LOGISTIC REGRESSION:** chosen for its popularity and logical approach



Comparing the results of the three classification models, Logistic Regression came out with the best result of 63% accuracy, followed by Support Vector Machine of 61%, while Naiv Bayes performed the least with 57% accuracy.

After several adjusments and manipulations made to have an improved output, all still came out with little or no effect, this leads to my conclusion that the independent variables chosen, inspite of their expected high impact and predictability on the target variable, they have very low relationship with the dependent variable, hence the reason for the not-so-impressive performance of the algorithms used.

**CONCLUSION**

**SUMMARISATION**

***Big Data Technologies:*** big data in its literal meaning is a lot of data, while the technology used in its processing, storage, mining, analysing, and visualization is the big data technology. Hosting these technologies and data infrastructures in the cloud gives them edge above traditional way of processing data.

In this project, I investigated some of the most popular and efficient big data warehouse platforms using some predefined criteria. I researched about how they operate, their infrastructure management, cost efficiency and ecosystem integration.

***EDA, classification and price prediction:*** To get insights from the given schema dataset, I used Pyspark, a general-purpose, in-memory, distributed processing engine in conjunction with Panadas and other important python libraries.

I created an argument where I was able to part the customers into two classes: spenders above and below average. I deployed some efficient machine learning models to make the predictions and I checked for the accuracies of these algorithms in comparison to each other.

The price prediction involved putting some high-impact features together, fed into ML algorithms to predict the price of product in a chosen organization, same as classification, I used different models and compared the results with other.

**EXPERIENCE GAINED**

In recent years, it has been confirmed that data generation has increased exponentially, hence traditional means of data processing is becoming obsolete. I got the opportunity to expose my self to better and more efficint means of processing data even on the petabyte level: Data warehousing.

I came to realise that the smaller the residual, the better the prediction, hence a ML model could be made better by ensuring a reduced residual i.e the gap between the target variable and the predicted variable

I also learnt that it is always important to feed the model with high-impact features, i.e features having strong correlation with the target variable, as these tend to increase the accuracy of the model.

**FUTURE WORK**

It is no doubt that data sources and types will continue to increase exponentially, it will be crucial to construct hyper-scale data warehouses with powerful computation and large storage capacity. Additionally, the incorporation of cloud computing technologies into data warehousing and the accessibility of big data solutions at reasonable prices cannot be overemphasized, this needs to be looked into as the technology progresses.

**REFERENCES**

*BigQuery Enterprise Data warehouse* (2019) *Google*. Google. Available at: https://cloud.google.com/bigquery (Accessed: January 7, 2023).

*Cloud computing services (AWS)* (2019) *Amazon web services.* Amazon. Available at: https://us-east-1.signin.aws.amazon.com/ (Accessed: January 5, 2023).

*Cloud computing services: Microsoft Azure* (2022) *Cloud Computing Services.*  Microsoft Azure. Available at: https://azure.microsoft.com/ (Accessed: January 7, 2023).

*Google BigQuery: A Serverless Data Warehouse* (no date) *Stitch*. Available at: https://www.stitchdata.com/resources/googlebigquery/#:~:text=In%20a%20serverless%20model%2C%20processing,and%20gaining%20insights%20from%20data. (Accessed: December 16, 2022).

*Introduction to optimizing query performance | Big Query | google cloud* (no date) *Google*. Google. Available at: https://cloud.google.com/bigquery/docs/best-practices-performance-overview (Accessed: December 16, 2022).

*Snowflake professional services: Optimize, accelerate, and achieve* (2022) *Snowflake*. Available at: https://www.snowflake.com/snowflake-professional-services/ (Accessed: January 7, 2023).

Team, K.P.I.P.N. (no date) *BigQuery best practices to optimize cost and performance*, *BigQuery Best Practices to Optimize Cost and Performance*. Available at: https://www.kpipartners.com/blog/bigquery-best-practices-to-optimize-cost-and-performance (Accessed: December 16, 2022).

Zhu, C.C. (2022) *Granular billing for Azure Data Factory*, *TECHCOMMUNITY.MICROSOFT.COM*. Available at: https://techcommunity.microsoft.com/t5/azure-data-factory-blog/granular-billing-for-azure-data-factory/ba-p/3654600 (Accessed: January 7, 2023).